



Open Archive TOULOUSE Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in : <http://oatao.univ-toulouse.fr/>
Eprints ID : 18849

The contribution was presented at CRE 2016 :
<http://cse2017.swc-rwth.de/>

To cite this version : Rocha Silva, Thiago and Winckler, Marco Antonio
Towards Automated Requirements Checking Throughout Development Processes of Interactive Systems. (2016) In: 2nd Workshop on Continuous Requirements Engineering (CRE 2016) co-located with the 22nd International Conference on Requirements Engineering: Foundation for Software Quality - REFSQ 2016, 14 March 2016 - 14 March 2016 (Gothenbrg, Sweden).

Any correspondence concerning this service should be sent to the repository administrator: staff-oatao@listes-diff.inp-toulouse.fr

Towards automated requirements checking throughout development processes of interactive systems

Thiago R. Silva, Marco A. A. Winckler

ICS-IRIT, Université Paul Sabatier
Toulouse, France

{rocha, winckler}@irit.fr

Abstract. The user-centered development process of interactive systems is iterative and, during multiple iterations, users have the opportunity to bring new requirements that are very likely to have an impact, not only in future development, but also affect previously developed artifacts. Manual testing of all artifacts when new requirements are introduced can be cumbersome and time consuming. For that, we need flexible methods to ensure continuous consistency and accuracy among the various artifacts employed to build interactive systems. The ultimate goal of this position paper is to briefly present our vision on an approach for automating the requirements assessment using a Behavior-Driven Development perspective. Thereby, automated tests can run early in the design process, providing a continuous quality assurance of requirements, and helping clients and teams to identify potential problems and inconsistencies before commitments with software implementation.

Keywords: Automated Requirements Checking, Behavior-Driven Development, Multi-Artifact Testing.

The activities of Requirements Engineering encompass cycles of meetings and interviews with clients, users and other stakeholders. The outcomes of these meetings are documented and modeled in several requirements artifacts which are cross-checked at the end of the process, when the system is functional and ready to be tested. When User-Centered Design (UCD) approaches are employed, intermediate tests can be conducted earlier in Prototypes. However, UCD practice has shown that users are keen to introduce new requirements along the process which might raise problems for the correspondence with artifacts already produced, demanding activities supported by Continuous Requirements Engineering. Currently, there are many solutions for tracing requirements specification artifacts such as Use Cases, Business Rules etc., but there is still room for investigating automated solutions for tracking and testing other types of artifacts such as Task Models, Prototypes etc. [2].

In this context, Behavior-Driven Development (BDD) [1] has aroused interest from both academic and industrial community in the last years. User Stories allow us to specify executable requirements in a natural language format, leading to a testable requirements specification and conducting to a “live” documentation, making easier

for clients receive feedback early and conduct their acceptance tests. For this reason, we are investigating some directions for using User Stories in a multi-artifact testing spectrum as follows.

From Ontologies to User Stories: Ontologies are useful to describe concepts used by platforms, models and artifacts that compose the design of interactive systems. We are studying ontologies for Web and Mobile platforms and associating the most common behaviors that each User Interface (UI) element can answer. These behaviors are being described using a natural language convention, useful later to specify steps of Scenarios in User Stories, in order to set actions on interaction elements.

From User Stories to Task Models: In Task Modeling, Scenarios represent instances of specific tasks described in the model. In User Stories, Scenarios describe examples of behaviors that users can perform in an interactive system. Ultimately, these behaviors can be viewed as examples of tasks being executed by users. Therefore, we are working on the use of User Stories as a mean to run Task Models, cross-checking them since early in the design process.

From User Stories to Prototypes: In a Requirements Engineering perspective, Prototypes are used to address opportunities for UI design in the early phases of development. This kind of artifact is highly evolutionary and can be useful until the last stages of development, when the Final UI is defined. Scenarios in the User Stories may support prototyping activities, giving them details and concrete examples of how performing tasks on the proposed interface. Considering this, we are using User Stories to run test scenarios in different stages of prototyping.

From User Stories to Final UIs: User Stories described in BDD are a powerful source of testing on the Final UI. Using external frameworks for specific environments, Scenarios can be directly executed, acting at the same time as executable requirements and test cases. We are writing User Stories from behaviors previously extracted from our ontologies. These behaviors are being tested by executable actions on the Final UI, using tools like JBehave and Selenium WebDriver.

These four directions highlight possible solutions to automate the requirements assessment, assuring VV&T¹ activities in a wider set of artifacts. In our approach, User Stories are central pieces for guiding multi-artifact testing. Moreover, the use of ontologies as a knowledge base to describe behaviors for UI elements allows us to provide multiple design solutions as well as promoting reuse when considering similar business models. Ongoing and future works include developing and adapting tools to visualize and run User Stories on the artifacts mentioned above, in addition to collect feedback from stakeholders through studies in real cases of software development.

References

1. Chelimsky, D., Astels, D., Helmkamp, B., North, D., Dennis, Z., & Hellesoy, A. (2010). The RSpec book: Behaviour driven development with Rspec, Cucumber, and friends. Pragmatic Bookshelf.
2. Nair, S., de la Vara, J. L., & Sen, S. (2013, July). A review of traceability research at the requirements engineering conference re@ 21. In Requirements Engineering Conference (RE), 2013 21st IEEE International (pp. 222-229). IEEE.

¹Validation, Verification and Testing